

IN THE CLAIMS

Please amend claims 1 and 11 as follows:

1. (Currently Amended) A computer-implemented method for processing rendering data containing vertices, comprising:
 - defining a vertex cache as a software cache located within a transform and lighting module and containing vertices in a floating-point format;
 - determining that a first vertex of the rendering data has already been transformed but not lighted and storing the first vertex in the vertex cache such that the first vertex bypasses a transformation module of the transform and lighting module;
 - converting the vertices in a floating-point format into a normalized homogeneous coordinate system (NHCS) fixed-point format by performing lighting and texture generation and transformation to obtain NHCS fixed-point format vertices;
 - performing view frustum clipping on the NHCS fixed-point format vertices after the lighting and texture generation and transformation;
 - transforming vertices of the rendering data that have not already been transformed from model space into clip space; and
 - continuing to store vertices of the rendering data that have already been transformed but not lighted in the vertex cache as needed to facilitate a single streamline branched architecture that avoids processing duplication of the vertices;
wherein defining a vertex cache, determining that a first vertex of the rendering data has already been transformed, converting the vertices, performing view frustum clipping, transforming vertices, and continuing to store vertices are performed on a processing unit of the computer.
2. (Original) The computer-implemented method of claim 1, further comprising examining each of the vertices before lighting to determine whether to cull.
3. (Original) The computer-implemented method of claim 2, further comprising discarding any vertices that are culled.

4. (Original) The computer-implemented method of claim 3, further comprising continuing processing of any vertices that are not culled.

5. (Canceled)

6. (Canceled)

7. (Canceled)

8. (Original) The computer-implemented method of claim 1, further comprising using Direct3D for mobile as a rendering standard.

9. (Canceled)

10. (Previously Presented) A computer-readable storage medium having stored and encoded thereon computer-executable instructions for performing on a computing device the computer-implemented method recited in claim 1.

11. (Currently Amended) A process for transforming and lighting rendering data, comprising:

inputting rendering data in model space containing vertices in a floating-point format;

converting the vertices in the floating-point format into a normalized homogeneous coordinate system (NHCS) fixed-point format;

transforming vertices in a NHCS fixed-point format in the rendering data from model space to clip space to generate transformed vertices;

culling a transformed vertex of the transformed vertices prior to processing by a lighting module after determining that the transformed vertex is not needed; and

lighting each of the transformed vertices using the lighting module, except for the culled transformed vertex, to compute color and generate transformed and lighted vertices from the rendering data;

wherein inputting rendering data, converting the vertices, transforming vertices, culling a transformed vertex, and lighting each of the transformed vertices are performed on a processing unit.

12. (Previously Presented) The process as set forth in claim 11, further comprising examining each of the transformed vertices to determine whether they have previously been lighted.

13. (Previously Presented) The process as set forth in claim 11, further comprising determining that the culled transformed vertex forms a back face of a triangle.

14. (Previously Presented) The process as set forth in claim 13, further comprising discarding the culled transformed vertex.

15. (Canceled)

16. (Previously Presented) The process as set forth in claim 11, further comprising determining that the culled transformed vertex is outside of one view frustum clip plane.

17. (Previously Presented) The process as set forth in claim 16, further comprising discarding the culled transformed vertex.

18. (Canceled)

19. (Previously Presented) One or more computer-readable storage media having stored and encoded thereon computer-readable instructions thereon which, when executed by one or more processors on a computing device, cause the one or more processors on the computing device to implement the process of claim 11.

20. (Previously Presented) A computer-implemented process stored and encoded on a computer-readable storage medium for rendering graphics on an embedded device, comprising:

inputting 3D data containing vertices in model space in a floating-point format;

converting the 3D data in a floating-point format into a normalized homogenous coordinate system (NHCS) fixed-point format in clip space to obtain NHCS fixed-point format vertices;

generating coordinates for the NHCS fixed-point format vertices by performing lighting and texture generation and transformation;

examining each of the NHCS fixed-point format vertices before lighting to determine whether to cull the NHCS fixed-point format vertices;

storing the NHCS fixed-point format vertices as needed in a vertex cache to provide a single streamline branched architecture that avoids processing duplication of the NHCS fixed-point format vertices; and

performing view frustum clipping of the NHCS fixed-point format vertices to generate an output of 2D screen coordinates to render the graphics represented by the rendering data on the embedded device.

21. (Previously Presented) The computer-implemented process of claim 20, wherein examining each of the NHCS fixed-point format vertices before lighting further comprises determining whether any of the NHCS fixed-point format vertices form a back face of a triangle and, if so, culling those NHCS fixed-point format vertices.

22. (Previously Presented) The computer-implemented process of claim 21, further comprising discarding each of the culled NHCS fixed-point format vertices and continuing to process NHCS fixed-point format vertices that have not been culled.

23. (Previously Presented) The computer-implemented process of claim 20, wherein examining each of the NHCS fixed-point format vertices before lighting further

comprises determining whether any of the NHCS fixed-point format vertices are outside of one view frustum clip plane and, if so, culling those NHCS fixed-point format vertices.

24. (Previously Presented) The computer-implemented process of claim 23, further comprising discarding each of the culled NHCS fixed-point format vertices and continuing to process NHCS fixed-point format vertices that have not been culled.

25. (Previously Presented) The computer-implemented process of claim 20, wherein the vertex cache is contained in software and not in hardware.

26. (Previously Presented) The computer-implemented process of claim 20, wherein the view frustum clipping is performed after a lighting and texture generation and transformation of the NHCS fixed-point format vertices.

27. (Canceled)

28. (Previously Amended) A transform and lighting module for preparing rendering data in a floating-point format for rendering, comprising:

an embedded computing device;

a computer-readable storage medium having stored and encoded thereon a computer program having program modules containing computer-executable instructions that are executable by the embedded computing device, the computer program further comprising:

a transformation module that converts vertices in a floating-point format in the rendering data into a normalized homogenous coordinate system (NHCS) fixed-point format in clip space to generate transformed vertices;

a vertex cache implemented as a software cache and located within the transform and lighting module that stores a first vertex contained in the rendering data such that the first vertex has previously been transformed but has not previously been lighted such that the first vertex is not processed by the transformation module;

a lighting module that computes color for each of the transformed vertices;

a culling module positioned after the transformation module and before the lighting module that culled a second vertex from the transformed vertices prior to processing by the lighting module after determining that the second vertex was not needed such that the second vertex is not processed by the lighting module;

a texture generation and texture transformation module that computes texture coordinates and transforms the texture coordinates into a fixed-point format in a normalized homogenous coordinate system (NHCS) to obtain NHCS fixed-point format vertices in clip space; and

a view frustum module positioned after the lighting module and after the texture generation and transformation module that performs view frustum clipping of the NHCS fixed-point format vertices in clip space to generate output data that can be rendered for display on a display device of the embedded computing device.

29. (Canceled)

30. (Original) The transformation and lighting module as set forth in claim 28, wherein the culling module performs at least one of: (a) back face culling; (b) view frustum culling.

31. (Canceled)